UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/694,568	10/27/2003	Stephen Michael Hartley	858-011568-US(PAR)	3544

2512 7590 08/30/2010

Perman & Green, LLP
99 Hawley Lane
Stratford, CT 06614

EXAMINER	
SALOMON, PHENUEL S	
ART UNIT	PAPER NUMBER
2179	
MAIL DATE	DELIVERY MODE
08/30/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/694,568	HARTLEY, STEPHEN MICHAEL	
	Examiner	Art Unit	
	PHENUEL S. SALOMON	2179	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 June 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-7,9-16,18-24,26-30,32-38,40-43,45-47,49-58,61 and 62* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-7,9-16,18-24,26-30,32-38,40-43,45-47,49-58,61 and 62* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the amendment filed on June 02, 2010. Claims 3, 8, 17, 25, 31, 39, 44, 48, 59, 60, and 63 are cancelled, and claims 1, 2, 4-7, 9-16, 18-24, 26-30, 32-38, 40-43, 45-47, 49-58, 61, and 62 are pending.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1, 2, 4-7, 9-16, 18-24, 26-30, 32-38, 40-43, 45-47, 49-58, 61, and 62 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bahrs (US 7,181,686 B1) in view of Hatanaka (US 5,926,177).

Claim 1: Bahrs discloses a method for routing views in a computer graphical user interface, comprising:

determining a view chain data structure comprising at least three entries, each of said entries comprising an application identifier and a view identifier, a view identified by said view identifier being within an application identified by said application identifier (*managing services in a desktop environment*) (col. 3, lines 15-17 and col. 16, lines 43-48);

passing said view chain data structure to a view router (*transporter 524*) from a first application (col. 3, lines 24-26, col. 16, lines 60-67 and col. 17, lines 1-5);

detecting a first entry in said view chain data structure by said view router (col. 3, lines 20-22, col. 16, lines 60-67 and col. 17, lines 1-5);

determining a first target application identifier in said first entry by said view router (col. 3, lines 22-24, col. 16, lines 60-67 and col. 17, lines 1-5);

launching a first view in said first target application by calling a method associated with said first target application, said launching comprising a presentation of a first user interface form in said computer graphical user interface by said first target application (col. 4, lines 22-30);

receiving data to the first view from a user during said launching of the first view (col. 4, lines 22-24);

continuing said view router by calling a listener method associated with said view router by said first target application (col. 3, lines 26-29).

Bahrs does not explicitly disclose:

checking whether entries for views not launched remain in said view chain data structure, each said entry for a view not launched specifying a view identifier for a view not yet launched. However, Hatanaka discloses the object also provides the interface and implementation for recording and supplying information about the real-world invocation that the object represents. This information includes, for instance, the state of the invocation (eg. Not started, starting, running, stopping, stopped, error or other internal states used in managing the invocation), the time it was started, any secondary invocations related to this object (eg. a data link from one

host to another will have a primary invocation on one host and a secondary invocation on another)…(col. 5, lines 1-15). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to include Hatanaka features in Bahrs. One would have been motivated to do so in order to quickly switching between views with no discrepancy between the data shown in the views (col. 4, lines 58-59).

Bahrs further discloses

detecting a second entry in said view chain data structure by said view router (col. 3, lines 20-22, col. 16, lines 60-67 and col. 17, lines 1-5) [in a desktop environment, this can also apply for a second entry];

determining a second target application identifier in said second entry by said view router (col. 3, lines 22-24, col. 16, lines 60-67 and col. 17, lines 1-5) [in a desktop environment, this can also apply for a second entry];

launching a second view in said second target application by calling a method associated with said second target application by said view router when entries for views not launched remain in said view chain, said launching comprising a presentation of a second user interface form in said computer graphical user interface by said second target application (col. 4, lines 22-30) [in a desktop environment, this can also apply for a second entry];

receiving data to the second view from the user during said launching of the second view (col. 4, lines 22-24) [in a desktop environment, this can also apply for a second entry];

continuing said view router by calling a listener method associated with said view router by said second target application (col. 3, lines 26-29) [in a desktop environment, this can also apply for a second entry];

continuing said first application automatically when no entries for views not launched remain in said view chain data structure by calling a listener method associated with said first application by said view router (col. 16, lines 22-35).

Claim 2. Bahrs and Hatanaka disclose the method according to claim 1, Bahrs further discloses the method further comprising:

gathering data from said first view and said second view (col. 4, lines 22-24); and passing said data from said view router to said first application or to a subsequent application identified in said view chain data structure (col. 4, lines 20-30).

Claim 4. Bahrs and Hatanaka disclose the method according to claim 2, Bahrs further discloses wherein said gathered data is organized into a journal list comprising an entry for each view in said view chain data structure (col. 19, lines 1-8).

Claim 5. Bahrs and Hatanaka disclose the method according to claim 2, Bahrs further discloses wherein said gathered data is organized into a list of type and value pairs (col. 48, lines 37-45).

Claim 6. Bahrs and Hatanaka disclose the method according to claim 5, Bahrs further discloses wherein said data type and value pair are defined in a markup language format (col. 48, line 37-45).

Claim 7. Bahrs and Hatanaka disclose the method according to claim 2, Bahrs further discloses wherein said view router provides a generic interface with generic methods and acts as an adapter for returning data from said first view to said first application or a subsequent application identified in said view chain data structure (col. 17, lines 1-6).

Claim 9. Bahrs and Hatanaka disclose the method according to claim 1, Bahrs further discloses wherein said view comprises user interface elements(fig. 5).

Claim 10. Bahrs and Hatanaka disclose the method according to claim 1, Bahrs further discloses wherein said view is a window opened during said launching step (fig. 5).

Claim 12. Bahrs and Hatanaka disclose the method according to claim 1, Bahrs further discloses wherein at least part of said view chain data structure is specified in a memory of an electronic device (fig. 2, item 232).

Claim 13. Bahrs and Hatanaka disclose the method according to claim 12, Bahrs further discloses wherein said view chain data structure is updated based on user actions (col. 4, lines 1-6).

Claim 14. Bahrs and Hatanaka disclose the method according to claim 1, Bahrs further discloses wherein said view chain data structure is determined based on user actions (col. 3, lines 42-49).

Claims 15, 29 and 43 respectively represent the system, the apparatus and the computer readable storage of the method claim 1 and are rejected along the same rationale.

Claims 16, 30 and 47 respectively represent the system, the apparatus and the computer readable storage of the method claim 2 and are rejected along the same rationale.

Claims 18, 32 and 49 respectively represent the system, the apparatus and the computer readable storage of method claim 4 and are rejected along the same rationale.

Claims 19, 33 and 50 respectively represent the system, the apparatus and the computer readable storage of method claim 5 and are rejected along the same rationale.

Claims 20, 34 and 51 respectively represent the system, the apparatus and the computer readable storage of method claim 6 and are rejected along the same rationale.

Claims 21, 35 and 52 respectively represent the system, the apparatus and the computer readable storage of method claim 7 and are rejected along the same rationale.

Claims 22, 36 and 53 respectively represent the system, the apparatus and the computer readable storage of method claim 8 and are rejected along the same rationale.

Claims 23, 37, and 54 respectively represent the system, the apparatus and the computer readable storage of method claim 9 and are rejected along the same rationale.

Claims 24, 38 and 55 respectively represent the system, the apparatus and the computer readable storage of method claim 10 and are rejected along the same rationale.

Claims 26, 40 and 56 respectively represent the system, the apparatus and the computer readable storage of method claim 12 and are rejected along the same rationale.

Claims 27, 41 and 57 respectively represent the system, the apparatus and the computer readable storage of method claim 13 and are rejected along the same rationale.

Claim 28, 42 and 58 respectively represent the system, the apparatus and the computer readable storage of method claim 14 and are rejected along the same rationale.

Claim 45. Bahrs and Hatanaka disclose the computer readable storage medium according to claim 43, Bahrs further discloses wherein said computer readable storage medium is a removable memory card (col. 66, lines 30-33).

Claim 46. Bahrs and Hatanaka disclose the computer readable storage medium according to claim 43, Bahrs further discloses wherein said computer readable storage medium is a magnetic or optical disk (col. 12, lines 61-63).

Claim 61. Bahrs and Hatanaka disclose the computer readable storage medium according to claim 43, Bahrs further discloses wherein said view router is implemented as a library (fig.26b).

Claim 62. Bahrs and Hatanaka disclose the computer readable storage medium according to claim 43, Bahrs further discloses wherein said view router is implemented as an own application (col. 5, lines 14-16).

Response to Arguments

4. Applicant's arguments filed on 6/2/2010 have been fully considered but are not persuasive.

Applicant argues Bahrs fails to disclose or suggest the feature of determining a view chain data structure comprising at least three entries, each of said entries comprising an application identifier and a view identifier, a view identified by said view identifier being within an application identified by said application identifier.

The examiner respectfully disagrees and notes that Bahrs discloses a desktop provides the operating-specific functionality of a windowing system and application management. For example, in Windows 95, the graphical interface that starts up is called a "desktop". The Windows 95 GUI itself is the "desktop system". Additionally, a desktop application provides a view or window in which an application may run and launch other applications. This is typically accomplished by a display of a hierarchical list of applications, which may be selected and

"launched" (col. 16, lines 40-48). On the desktop, there is a view chain of hierarchical applications where selection of any view will cause the program execution. Bahrs further discloses a diagram illustrating runtime behavior for a TopListener subsystem is depicted in accordance with a preferred embodiment of the present invention. The runtime behavior of the TopListener subsystem 5700 is shown in FIG. 57. TopListener 5702 performs generic operations on the business desktop, as specified by a TopEvent 5704. The TopEvent and TopListener have several functions. One function is interaction with the host environment (i.e. for a Java application--the operating system), with the host environment services (i.e. other applications on the operating sytem), with the application enabler (i.e. the Netscape browser and its status line or maybe aspects of the Java Virtual Machine), and with host environment policies (i.e. how is shutdown handled, how are error messages handled). The hosting client desktop environment is an embodiment of one or more of these items (col. 41, lines. 32-47).

Applicant further argues Bahrs fails to disclose passing said view chain data structure to a view router from a first application.

The examiner respectfully disagrees and notes that Bahrs discloses wherein the view controller handles user input to the graphical user interface. Responsive to a selected user input, the selected user input is sent from the view controller to an application mediator. Responsive to receiving the selected user input at the application mediator, the selected user input is processed at the application mediator. Responsive to the application mediator determining that a service is

required in the desktop environment, an event is generated (col. 3, lines 19-29). Furthermore, Bahrs discloses a *transporter* which considers as a *view router* that works in close coordination with the application mediator to process the views: (*....transporter 524 will route RequestEvents, such as RequestEvent 526 to various Destinations, such as Destination 528, Destination 530, or Destination 532. These destinations will interpret the RequestEvent 526, locate a server, create a message format, create a protocol and deliver the information to the server's service for processing. These response data will be returned to the Transporter 524 in a RequestEvent, such as RequestEvent 526. In turn, Transporter 524 will return the RequestEvent to ApplicationMediator 512, which will process the data contained in the event accordingly. For example, the return data may be sent to ViewController 502 to refresh the view displayed on the screen to the user. If an error occurs, a RequestException 534 may be generated and returned to Transporter 524 and returned to the ApplicationMediator 512*) (col. 16, lines 60-67 and col. 17, lines 1-5). Bahrs clearly discloses passing a view chain data structure to a view router from a first application. Bahrs further discloses it is possible to build a chain of validation rules. FIG. 47 illustrates the application of two edit rules, range checking (that is, the value is within specific limits) and formatting for viewing. The separate rules are assigned to string values named range and money. This string values are then put into an array of strings. The transmittable form of the data is stored in value. The method applyEdits applies both validation rules to this data; if any exception occurs, it is handled by the catch portion of the try statement. The setText method from the Java AWT is used to redisplay the formatted data.(col. 38, lines 62-67 and col. 39, lines 1-4), (col. 59, lines 20-43).

Applicant further argues Bahrs fails to disclose detecting a first entry in said view chain data structure by said view router.

The examiner respectfully disagrees and notes that Bahrs discloses responsive to a selected user input, the selected user input is sent from the view controller to an application mediator. Responsive to receiving the selected user input at the application mediator, the selected user input is processed at the application mediator (col. 3, lines 19-29). Furthermore, Bahrs discloses a *transporter* which considers as a *view router* that works in close coordination with the application mediator to process the views: (*….transporter 524 will route RequestEvents, such as RequestEvent 526 to various Destinations, such as Destination 528, Destination 530, or Destination 532. These destinations will interpret the RequestEvent 526, locate a server, create a message format, create a protocol and deliver the information to the server's service for processing. These response data will be returned to the Transporter 524 in a RequestEvent, such as RequestEvent 526. In turn, Transporter 524 will return the RequestEvent to ApplicationMediator 512, which will process the data contained in the event accordingly. For example, the return data may be sent to ViewController 502 to refresh the view displayed on the screen to the user. If an error occurs, a RequestException 534 may be generated and returned to Transporter 524 and returned to the ApplicationMediator 512*) (col. 16, lines 60-67 and col. 17, lines 1-5).

Applicant argues Bahrs fails to disclose determining a first target application identifier in said first entry by said view router.

The examiner respectfully disagrees and notes that Bahrs discloses …additionally, a desktop application provides a view or window in which an application may run and launch other applications. This is typically accomplished by a display of a hierarchical list of applications, which may be selected and "launched" (col. 16, lines 40-48) and Bahrs further discloses (*….transporter 524 will route RequestEvents, such as RequestEvent 526 to various Destinations, such as Destination 528, Destination 530, or Destination 532. These destinations will interpret the RequestEvent 526, locate a server, create a message format, create a protocol and deliver the information to the server's service for processing. These response data will be returned to the Transporter 524 in a RequestEvent, such as RequestEvent 526. In turn, Transporter 524 will return the RequestEvent to ApplicationMediator 512, which will process the data contained in the event accordingly. For example, the return data may be sent to ViewController 502 to refresh the view displayed on the screen to the user. If an error occurs, a RequestException 534 may be generated and returned to Transporter 524 and returned to the ApplicationMediator 512*) (col. 16, lines 60-67 and col. 17, lines 1-5).

Applicant argues Bahrs fails to disclose launching a first view in said first target application by calling a method associated with said first target application, said launching

comprising a presentation of a first user interface form in said computer graphical user interface by said first target application.

The examiner respectfully disagrees and notes that Bahrs discloses a graphical user interface is presented using a view controller, wherein the view controller handles the user input to the graphical user interface. Responsive to a selected user input, an event is sent to a first application mediator. Responsive to the first application mediator being unable to process the event, the event is sent to a second application mediator for processing, wherein the first application mediator and the second application mediator handle an order in which a set of displays are displayed by a view controller (col. 4, lines 22-30). And Bahrs further discloses additionally, a desktop application provides a view or window in which an application may run and launch other applications. This is typically accomplished by a display of a hierarchical list of applications, which may be selected and "launched" (col. 16, lines 40-48).

Applicant further argues Bahrs fails to disclose launching a second view in said second target application by calling a method associated with said second target application by said view router when entries for views not launched remain in said view chain, said launching comprising a presentation of a second user interface form in said computer graphical user interface by said second target application

The examiner respectfully disagrees and notes that Bahrs discloses launching a second view in said second target application by calling a method associated with said second target application by said view router (col. 4, lines 22-30) [in a desktop environment, this can also apply for a second entry] and when entries for views not launched remain in said view chain, said launching comprising a presentation of a second user interface form in said computer graphical user interface by said second target application (col. 51, lines 61-67).

Applicant argues there is also no disclosure or suggestion in Bahrs related to continuing said first application automatically when no entries for views not launched remain in said view chain data structure by calling a listener method associated with said first application by said view router.

The examiner respectfully disagrees and notes that Bahrs discloses the process begins by selecting a ViewListener (step 8800) that has not yet been processed from the list of ViewListeners that have been added to using addViewListener. Thereafter, a determination is made as to whether ViewListeners are present for processing (step 8802). If ViewListeners are present, then invoke the method ViewListener(i).viewEventPerformed (step 8804). This method is implemented by a ViewListener (such as an ApplicationMediator) to process the ViewEvent passed. Typically, this processing of the ViewEvent will occur on a separate thread. Also, the ViewListener may decide to "consume" the event by calling the consume( ) method on the

ViewEvent. This essentially specifies that the ViewEvent should not be sent to any other ViewListeners left to be processed (col. 51, lines 45-67).

Applicant argues Hatanaka fails to disclose the feature of checking whether entries for views not launched remain in said view chain data structure, each said entry for a view not launched specifying a view identifier for a view not yet launched.

The examiner respectfully disagrees and notes that Hatanaka discloses each of the views has an associated display; functional view 106 has a functional view display (icon view) 107, real view 108 has a real view display (list view) 109, and the Nth view 110 has an Nth view display 111. The Controller 100 has a user input 114 for initializing and providing changes to the views. The Controller 100 provides view management at 120 to the ViewProxy and model management at 122 to the Model 102 (col. 3, lines 30-36). Hatanaka further discloses this information includes, for instance, the state of the invocation (eg. Not started, starting, running, stopping, stopped, error or other internal states used in managing the invocation) (col. 5, lines 3-6). For example, (Not started, starting, running, stopping, stopped, error or other internal states used in managing the invocation) all those processes are related to view presentation where different states are being determined in order to process a particular view.

Conclusion

5. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a). A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. Hyndman et al. (US 6,161,136).

b. Weitzman (US 2003/0197726 A1).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phenuel S. Salomon whose telephone number is (571) 270-1699. The examiner can normally be reached on Mon-Fri 7:00 A.M. to 4:00 P.M. (Alternate Friday Off) EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on (571) 272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-3800.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Phenuel S. Salomon/
Examiner, Art Unit 2179

/Steven B Theriault/

Primary Examiner, Art Unit 2179